

Non solo Django: MVC orientato agli oggetti con Plone e Zope Toolkit

Maurizio Delmonte - Abstract

MVC?.. ah, MVC!

Model - View - Controller [1], stiamo parlando di questo:

- architettare l'applicazione separando la gestione del dato (model), dalla sua presentazione (view) e dall'interazione da parte dell'utente (controller)
- Django e molti altri framework ci hanno basato il loro successo (le applicazioni risultano *naturalmente* più robuste, e facili da mantenere e far evolvere)

[1]: cfr. <http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>

quale problema vogliamo risolvere?

Titolo Home

Descrizione Home

Progetto:
- titolo
- immagine
- descrizione

Lista Post:
- data
- titolo
- descrizione
- immagine

Nikko Bautista | Web Application Developer
http://nikkobautista.com/

Nikko Bautista

Web Application Developer

Home | [About](#) | [Projects](#) | [Blog](#) | [Contact](#)

Lorem Ipsum Dolor Sit Amet

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

Featured Project

Image

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. [Read More >](#)

From the Blog

[Lorem Ipsum Dolor Sit Amet](#)
Pellentesque eleifend posuere eros, luctus pretium nulla adipiscing ac. Aenean blandit feugiat augue, sed gravida libero malesuada eu. Cras fermentum nisl et dui lacinia ultrices. Curabitur nec turpis at est molestie sagittis in ut purus. [Read More >](#)

[Morbi in Sem Quis Dui Placerat Ornare](#)
Gravida libero malesuada eu. Cras fermentum nisl et dui lacinia ultrices. Curabitur nec turpis at est molestie sagittis in ut purus. Lorem ipsum dolor sit amet, pede morbi consectetur porttitor nibh, in consectetur in sit ac lorem. [Read More >](#)

[Pellentesque Odio Nisi Euismod In](#)
Eleifend posuere eros, luctus pretium nulla adipiscing ac. Aenean blandit feugiat augue, sed gravida libero malesuada eu. Cras fermentum nisl et dui lacinia ultrices. Curabitur nec turpis at est molestie sagittis in ut purus. [Read More >](#)

[See My Projects](#)

[Visit My Blog](#) or [Subscribe via RSS](#)

Twitter
Donec eu libero sit amet quam egestas semper. Aenean

My Writing Elsewhere
[A Beginner's Guide to Design Patterns](#)
Ever wondered what design patterns are? In

Copyright (C) 2010 by Nikko Bautista

MVC, l'interpretazione Zope (2)

- **Model** -> isolare la gestione del dato con ZODB
- **View** -> costruire l'interfaccia utente con Template ZPT e Browser View
- **Controller** -> applicare le viste ai modelli tramite il Traversing

NB: Zope (2) != Zope [cfr. <http://zope2.zope.org/> e <http://www.zope.org/the-world-of-zope>]

Mvc -> ZODB

- **DB a oggetti** (NO-SQL *ante litteram dal 1998*): DB per oggetti Python
- efficiente e robusto (transazionale **ACID**o q.b.)
- gestione *trasparente* dei meccanismi CRUD
- Storage pluggabile
- utile *fuori* da Zope

approfondite qui: <http://zodb.org/>

mVc-> ZPT e Browser view

- **ZPT** (Zope Page Templates) - linguaggio per i template realizzato per Zope (dal **2000**)
- orientato alla produzione di XML e XHTML
- "poco" potente, per non favorire chi pretende di infilare le logiche di controllo nei template

approfondite qui: <http://wiki.zope.org/ZPT/FrontPage>

- **Browser View** (ZPT + classe python: *componente* secondo Zope Toolkit) -> ne parliamo più avanti..

mvC -> Z-Publisher e Traversing

Lo Z-Publisher dal 1998 si occupa di:

- Gestione di REQUEST e RESPONSE
- Gestione degli argomenti della REQUEST
- **Traversing**: ogni URL al suo oggetto
- imposta autenticazione utente
- effettua la **call** dell'oggetto/metodo risultato del traversing, con gli argomenti della REQUEST

approfondite qui: <http://wiki.zope.org/zope2/ZPublisher>

ricapitolando..

URL: *http://localhost:8080/parent/child/method?arg=test*

tipicamente:

- a. **method** è un *metodo pubblico* dell'oggetto **child** memorizzato in ZODB come figlio dell'oggetto **parent**
- b. **method** è un oggetto memorizzato in ZODB (es. un template ZPT)

Lo Z-Publisher **risolve** *method*, e lo **invoca** passandogli il parametro *arg=test*.

ZOPE -> Z-Object Publishing Environment

i componenti di Zope Toolkit

Zope Toolkit (**2005**) è una libreria inclusa in Zope 2 da qualche anno, prima nota come *Zope 3*.

Il cuore di ZTK è la ZCA, che abilita il mondo dei **componenti** nelle applicazioni Zope.

approfondite qui:

<http://docs.zope.org/zopetoolkit/>

<http://wiki.zope.org/zope3/ComponentArchitectureOverview>

Browser view

- componenti "Adapter" dello ZTK
- implementazione: zpt + py + zcml

Z-Publisher assume una nuova dimensione:

URL: *http://localhost:8080/parent/**child/method**?arg=test*

con ZTK:

.. c. **method** è una Browser View

NB: non vive nello ZODB.

come si arriva a Plone

I concetti introdotti sono alla base delle applicazioni Zope 2.

Plone è un'applicazione Zope 2 centrata sui contenuti:

- gli oggetti di Plone rappresentano **contenuti**
- gli oggetti di Plone implementano **servizi**
 - per gestire contenuti
 - per costruire l'interfaccia utente

Content Framework

Gestire contenuti in agilità: **content type**.

- factory e memorizzazione
- schema di attributi
- metadati
- configurazione
- registrazioni ai servizi (indicizzazione, workflow, trasformazione, etc.)

tutto questo avviene in modo omogeneo, coerente e robusto.

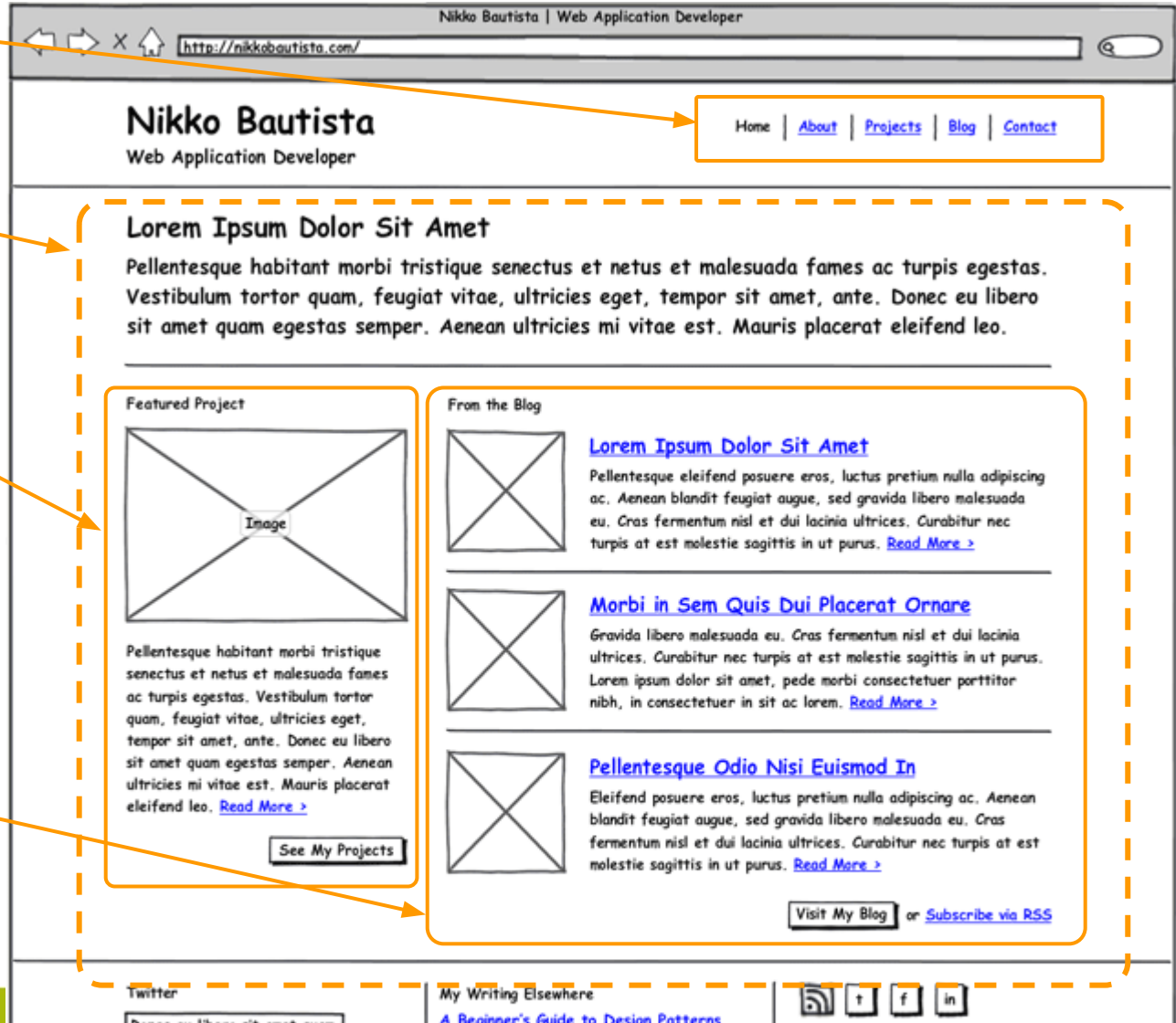
risolvere con i content types..

Cartelle, Pagine, Progetti, Post

Portlet Page: titolo, descrizione e due portlet collection.

Progetto:
 - titolo
 - immagine
 - descrizione

Post:
 - data
 - titolo
 - descrizione
 - immagine



Archetypes e Dexterity

Plone ha due Content Framework:

- **Archetypes** (dal **2003**) - monolitico, è il cuore di Plone da 10 anni
- **Dexterity** (dal **2010**) - a componenti, sarà incluso in Plone 4.3 base

MVC con Content Framework

- Model - **automatico** da Content Framework
- View - di base **automatico** da Content Framework
- Controller - di base **automatico** da Content Framework

MVC automatizzato in Plone

Lo sviluppatore definisce un content type dichiarando:

- **Factory Type Information** (DNA del content type)
- **schema** di attributi

In *automatico* il content framework gestisce:

- interfaccia utente base di view/edit
- factory e logiche di aggiornamento
- interazione con i servizi in base a configurazioni (indicizzazione, workflow, etc.)

ricapitolando..

URL: *http://localhost:8080/page*

http://localhost:8080/page/edit

- **page** è un oggetto *contenuto* memorizzato in ZODB
- *page* ha una *vista di default* invocata automaticamente e restituita all'utente
- **edit** è un "metodo applicato al contesto" dell'oggetto *page* (View e Controller insieme)

NB: queste viste sono costruite automaticamente dal content framework.

demo.. *statica*

vediamo in rapida successione:

- una pagina plone da utente anonimo
- la stessa pagina dopo il login
- la stessa pagina in vista edit

da notare:

- le varie funzioni attivabili da UI
- la vista contents sulla cartella

accesso "anonimo" alle pagine

<http://127.0.0.1:8080/Plone>



Fatti riconoscere

Cerca nel sito
 solo nella sezione corrente



Tu sei qui: Home

Benvenuto in Plone

Congratulazioni! Hai completato con successo l'installazione di Plone.

[Disponibile anche in modalità presentazione...](#)

Se stai visualizzando questa pagina, invece del sito web che ti aspettavi, è perché il gestore di questo sito ha appena installato Plone. Non contattare il Team o la mailing list di Plone per questo.

Prime operazioni

Prima di cominciare a esplorare il tuo nuovo sito Plone, esegui le seguenti operazioni:

1. Assicurati di essere collegato come utente amministratore/manager (dovresti vedere un link alla configurazione del sito nell'angolo in alto a destra)
2. [Configura il server di posta elettronica](#) (Plone necessita di un server SMTP valido per verificare l'utente e spedire le istruzioni per reimpostare la propria password quando viene smarrita)
3. [Decidi che livello di sicurezza applicare](#) (chi può iscriversi, modalità assegnazione password, ecc.)

Mettiti comodo

accesso editor alle pagine

<http://127.0.0.1:8080/Plone>



admin ▼

Cerca nel sito Cerca

solo nella sezione corrente



Tu sei qui: Home



Benvenuto in Plone

creato da [admin](#) — ultima modifica 12/06/2012 19:16 — [Cronologia](#)

Congratulazioni! Hai completato con successo l'installazione di Plone.

[Disponibile anche in modalità presentazione...](#)

Se stai visualizzando questa pagina, invece del sito web che ti aspettavi, è perché il gestore di questo sito ha appena installato Plone. Non contattare il Team o la mailing list di Plone per questo.

Prime operazioni

Prima di cominciare a esplorare il tuo nuovo sito Plone, esegui le seguenti operazioni:

1. **Assicurati di essere collegato come utente amministratore/manager** (dovresti vedere un link alla configurazione del sito nell'angolo in alto a destra)
2. **Configura il server di posta elettronica** (Plone necessita di un server SMTP valido per verificare l'utente e spedire le istruzioni per reimpostare la propria password quando viene smarrita)
3. **Decidi che livello di sicurezza applicare** (chi può iscriversi, modalità assegnazione password, ecc.)

modifica della pagina

Contenuti Visualizza **Modifica** Regole Condivisione

Modifica Pagina

<http://127.0.0.1:8080/Plone/front-page/edit>

Informazioni Stai modificando la vista di default del contenitore. Se vuoi invece modificare il contenitore stesso, [vai qui](#).

Default ■ Categorizzazione Date Proprietario Impostazioni

Titolo ■

Benvenuto in Plone

Descrizione
Usato nell'elenco degli elementi e nei risultati delle ricerche.

Congratulazioni! Hai completato con successo l'installazione di Plone.

Testo del documento

Stili... **B** *I* [List icons]

Se stai visualizzando questa pagina, invece del sito web che ti aspettavi, è perché il gestore di questo sito ha appena installato Plone. Non contattare il Team o la list di Plone per questo

Prime operazioni

Prima di cominciare a esplorare il tuo nuovo sito Plone, esegui le seguenti operazioni:

..modifica alla pagina II

Default ■ **Categorizzazione** Date Proprietario Impostazioni

Categorie

Le categorie, conosciute anche come parole chiave o etichette (tag), sono c

Crea ed aggiungi nuove categorie

Inserisci una categoria per linea, puoi inserire categorie di più parole su una linea.

Elementi correlati

Aggiungi...

Località:

La locazione geografica dell'oggetto, se applicabile.

Lingua

Italian ▾

Default ■ **Categorizzazione** **Date** Proprietario Impostazioni

Data di pubblicazione

La data in cui l'elemento verrà pubblicato. Se non viene inserita alcuna data l'e

-- ▾ / -- ▾ / -- ▾  -- ▾ : -- ▾

Data di scadenza

La data di scadenza del documento. Dopo questa data, il documento sarà resc viene inserita alcuna data il documento rimarrà visibile indefinitamente.

-- ▾ / -- ▾ / -- ▾  -- ▾ : -- ▾

Default ■ **Categorizzazione** Date **Impostazioni** Proprietario

Consenti i commenti

Se selezionato, gli utenti potranno aggiungere dei commenti a questo elemento.

Escludi dalla navigazione

Se selezionato, questo elemento non apparirà nell'albero di navigazione

Modalità di presentazione

Se selezionato, gli utenti avranno la possibilità di vedere i contenuti come sequenza di diapositive.

Tabella dei contenuti

Se selezionato, verrà mostrato un indice dei contenuti in cima alla pagina.

e per le cartelle.. folder contents

http://127.0.0.1:8080/Plone/folder_contents

admin ▾



Cerca nel sito Cerca
 solo nella sezione corrente

Home Notizie Eventi Collaboratori

Tu sei qui: Home

Contenuti Visualizza Regole Condivisione Aggiungi... ▾

- Cartella
- Collegamento
- Collezione
- Evento
- File
- Immagine
- Notizia
- Pagina

Sito

ultima modifica 12/06/2012 19:16

Selezione: Tutti

		Titolo	Dimensione	Modificato	Stato
⋮	<input type="checkbox"/>	Benvenuto in Plone ■	5.1 kB	12/06/2012 19:16	Pubblicato
⋮	<input type="checkbox"/>	Notizie	1 kB	12/06/2012 19:16	Pubblicato
⋮	<input type="checkbox"/>	Eventi	1 kB	12/06/2012 19:16	Pubblicato
⋮	<input type="checkbox"/>	Collaboratori	1 kB	12/06/2012 19:16	Pubblicato

Copia Taglia Rinomina Elimina Cambia lo stato



Ricapitolando..

Portale => gerarchia di oggetti (**content types**), organizzati in cartelle.

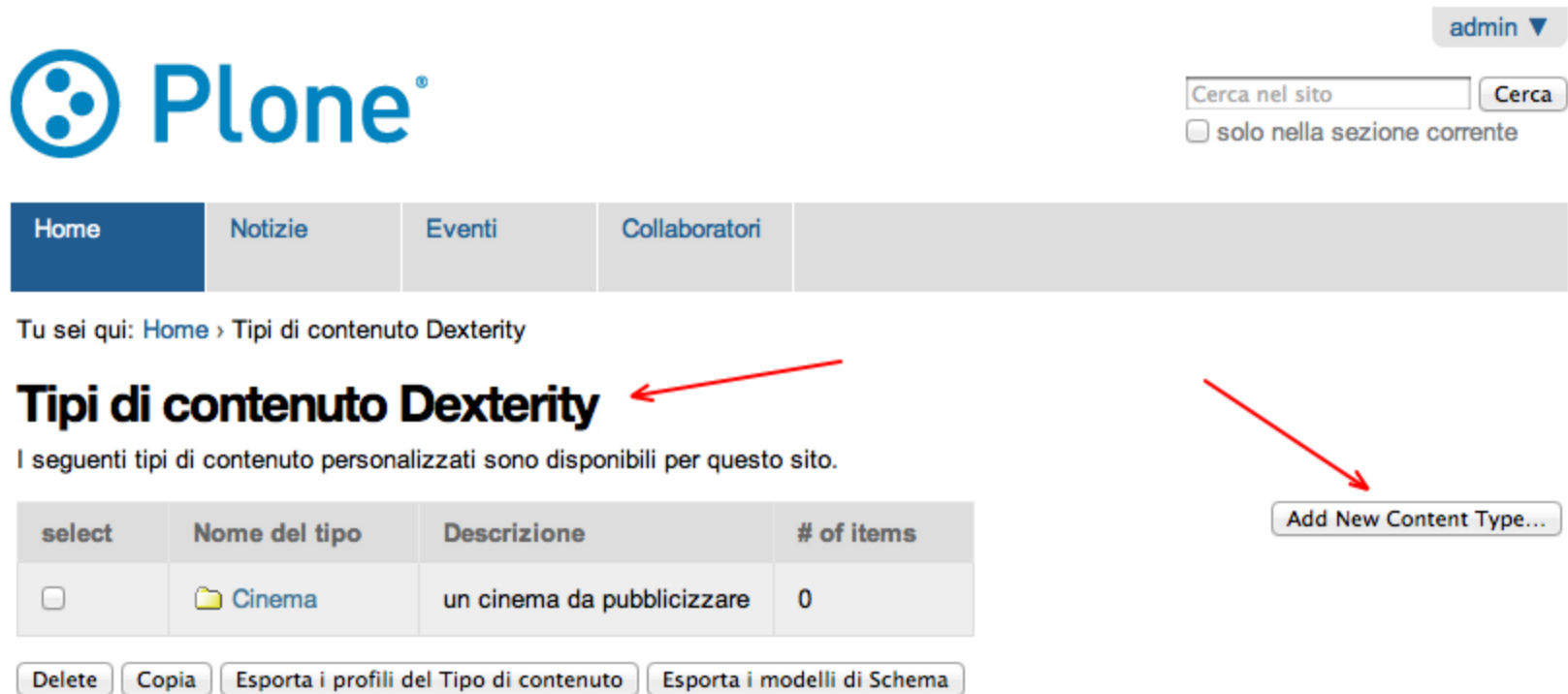
URL => posizione fisica dell'oggetto + vista applicata

M => oggetto memorizzato in ZODB

VC => template ZPT / browser view associati ai content type

il content framework gestisce
MVC automaticamente.

Content Type via web con Dexterity




The screenshot shows the Plone administration interface. At the top right, there is a user menu for 'admin' and a search bar with the text 'Cerca nel sito' and a 'Cerca' button. Below the search bar is a checkbox labeled 'solo nella sezione corrente'. A navigation menu at the top contains 'Home', 'Notizie', 'Eventi', and 'Collaboratori'. The main content area shows a breadcrumb trail 'Tu sei qui: Home > Tipi di contenuto Dexterity' followed by the heading 'Tipi di contenuto Dexterity'. Below the heading is a table of content types. A red arrow points from the heading to the table, and another red arrow points from the heading to an 'Add New Content Type...' button on the right. Below the table are buttons for 'Delete', 'Copia', 'Esporta i profili del Tipo di contenuto', and 'Esporta i modelli di Schema'.

Tu sei qui: [Home](#) > Tipi di contenuto Dexterity

Tipi di contenuto Dexterity

I seguenti tipi di contenuto personalizzati sono disponibili per questo sito.

select	Nome del tipo	Descrizione	# of items
<input type="checkbox"/>	 Cinema	un cinema da pubblicizzare	0

[Delete](#)
[Copia](#)
[Esporta i profili del Tipo di contenuto](#)
[Esporta i modelli di Schema](#)

[Add New Content Type...](#)

NB: installare **plone.app.dexterity** [*], incluso in Plone 4.3.
 [*]: <http://pypi.python.org/pypi/plone.app.dexterity>

gestione dello schema attributi

« Back to Tipi di contenuto Dexterity

Fields Behaviors

Modifica Cinema (cinema)

Aggiungi un nuovo campo...

Default

<p>☰ nome – Linea di testo (Stringa)</p> <p>Nome ■</p> <input type="text"/>	<p>Impostazioni... ✕</p>
<p>☰ numero_sale – Intero</p> <p>Numero Sale ■</p> <input type="text"/>	<p>Impostazioni... ✕</p>

Save Defaults

molte delle caratteristiche dello *schema attributi* possono essere gestite tramite browser.

aggiunta di un campo


 solo nella sezione corrente

- Home
- Notizie
- Eventi
- Collaboratori

Tu sei qui: [Home](#) > [Tipi di contenuto Dexterity](#) > [Cinema](#)

[« Back to Tipi di contenuto Dexterity](#)

Fields Behaviors

Modifica Cinema

Default

- nome – Linea di testo
- numero_sale – Intero

Numero Sale

Save Defaults

Aggiungi un nuovo campo

Titolo ■

Nome breve ■
Utilizzato per l'accesso programmatico al campo.

Testo di aiuto
Si presenta sotto forma di testo di aiuto per il campo.

Tipo di campo ■



Dexterity in azione

Aggiungi Cinema

un cinema da pubblicizzare

Contenuto ■ Categorization Dates Ownership

Titolo ■

Descrizione

Usato nell'elenco degli elementi e nei risultati delle ricerche.

Nome ■

Numero Sale ■

Orario Spettacoli ■

Salva

Annulla

.. ma non vogliamo il Dublin Core!

i behavior di Dexterity

[« Back to Tipi di contenuto Dexterity](#)

Fields

Behaviors

Behaviors for Cinema (cinema)

Metadati di base

Aggiunge un titolo e una descrizione

Categorizzazione

Aggiunge le categorie e l'impostazione della lingua

Metadati Dublin Core

Aggiunge i campi dei metadati Dublin Core (uguale a Metadati di base + Categorizzazione + Durata effettiva + Possessore)

Durata effettiva

Assegna una data di pubblicazione e/o di scadenza

Escludi dalla navigazione

Allow items to be excluded from navigation

Nome dal nome del file

Generazione automatica del nome breve URL per il contenuto in base al nome del suo campo primario

Nome dal titolo

Generazione automatica del nome breve URL per il contenuto in base al suo titolo iniziale

Radice di navigazione

Make all items of this type a navigation root. (Warning: this affects site navigation and is intended for content types that act as their own subsite.)

Navigazione precedente successivo

Dexterity in azione 2

Aggiungi Cinema

un cinema da pubblicizzare

Nome ■

Numero Sale ■

Orario Spettacoli ■

Salva

Annulla

cosa sono i **behavior**?

Adapter (logiche python registrate nella ZCA) attivati sui content type da specifiche interfacce marker.

cosa fare con i **behavior**?

- attivare schemi di attributi
- attivare logiche speciali
- accedere a funzioni aggiunte da estensioni
- ...

Sìmo dice:

Behaviors?

"Generic Programming"!

[cfr. http://en.wikipedia.org/wiki/Generic_programming]

NB: i behaviour sono "indipendenti" e "ortogonali" ai content type.

gestire le Factory Type Informations

Dexterity FTI at /Plone/portal_types/cinema ←

Properties allow you to assign simple values to Zope objects. To change pr

Name	Value
Title	Cinema
Description	un cinema da pubblicizzare
I18n Domain	plone
Icon (Expression)	string:\${portal_url}/folder_icon.i
Product factory	cinema
Add view URL (Expression)	string:\${folder_url}/++add++ci
Add view link target	
Initial view name	view
Implicitly addable?	<input checked="" type="checkbox"/>
Filter content types?	<input type="checkbox"/>
Allowed content types	ATBooleanCriterion ATCurrentAuthorCriterion ATDateCriteria ATDateRangeCriterion ATListCriterion ATPathCriterion ATPortalTypeCriterion
Allow Discussion?	<input type="checkbox"/>
Default view method	view
Available view methods	view

Alcune configurazioni di registro *resistono* nella ZMI

In portal_types è possibile gestire i portal type del portale:

- controllando alcune configurazioni
- controllando le viste associate
- creando nuovi content types senza passare dal codice

Dexterity in un pacchetto Python



admin ▼

Cerca nel sito

Cerca

 solo nella sezione corrente

Home

Notizie


Eventi

Collaboratori

Tu sei qui: [Home](#) > [Tipi di contenuto Dexterity](#)

Tipi di contenuto Dexterity

I seguenti tipi di contenuto personalizzati sono disponibili per questo sito.

select	Nome del tipo	Descrizione	# of items
<input type="checkbox"/>	 Cinema	un cinema da pubblicizzare	0

Delete

Copia

Esporta i profili del Tipo di contenuto

Esporta i modelli di Schema

Dal registro dei Content Type Dexterity si può:

- esportare le FTI
- esportare gli schemi di attributi

approfondite qui: <http://plone.org/products/dexterity/documentation/manual/developer-manual>

export dello schema Dexterity

```
<model xmlns="http://namespaces.plone.org/supermodel/schema">
  <schema>
    <field name="nome"
      type="zope.schema.TextLine">
      <title>Nome</title>
    </field>
    <field name="numero_sale"
      type="zope.schema.Int">
      <title>Numero Sale</title>
    </field>
    <field name="orario_spettacoli"
      type="zope.schema.Text">
      <title>Orario Spettacoli</title>
    </field>
  </schema>
</model>
```

NB: lo stesso schema può essere espresso in maniera *pythonica* con una speciale interfaccia di ZTK.

```
from plone.directives import form
from zope import import schema
```

```
class ICinema(form.Schema):
    "Un Cinema."
    nome = schema.TextLine(
        title=u'Nome')
    numero_sale = schema.Int(
        title=u'Numero Sale')
    orario_spettacoli = schema.Text(
        title=u'Orario Spettacoli')
```

export del FTI Dexterity

```
<?xml version="1.0"?>
<object name="cinema" meta_type="Dexterity FTI" i18n:domain="plone"
  xmlns:i18n="http://xml.zope.org/namespaces/i18n">
  <property name="title" i18n:translate="">Cinema</property>
  <property name="factory">cinema</property>
  <property name="add_view_expr">string:${folder_url}/++add++cinema</property>
  <property name="default_view">view</property>
  <property name="klass">plone.dexterity.content.Container</property>
  <property name="behaviors">
    <element value="plone.app.content.interfaces.INameFromTitle"/>
  </property>
  <property name="schema"> ... </property>
  ...
</object>
```

NB: la FTI XML viene presa in carico dal registro dei content types tramite importazione standard da portal_setup

dove sta il Python?

Niente Panico!

L'XML che avete visto è costruito *"dalla macchina per la macchina"*.

Chi sviluppa Dexterity pensa che le configurazioni stanno bene vestite di XML.

Chi sviluppa **con Dexterity** non ha bisogno di manipolare XML, se non ha piacere a farlo.

Documentazione Dexterity!

Dexterity è molto ben documentato!!

- **FAQ [1]**
- **Dexterity Developer Manual [2]**
- **Creare behavior riusabili per content type Dexterity [3]**

[1]: <http://plone.org/products/dexterity/documentation/faq>

[2]: <http://plone.org/products/dexterity/documentation/manual/developer-manual>

[3]: <http://plone.org/products/dexterity/documentation/manual/behaviors>

realizzare una nuova vista

```
from five import grok
from Acquisition import aq_inner
from Products.CMFCore.utils import getToolByName
```

nome vista: **@@view**
(nome classe lowercase)

```
class View(grok.View):
    grok.context(ICinema)
    grok.require('zope2.View')
```

vista applicata a oggetti
con interfaccia **ICinema**

ammessi solo utenti con
permesso **zope2.View**

```
def lista_cinema(self):
    "lista di tutti i cinema disponibili"
    context = aq_inner(self.context)
    catalog = getToolByName(context, 'portal_catalog')
    return catalog(object_provides=ICinema.__identifier__,
                   path='/'.join(context.getPhysicalPath()),
                   sort_on='sortable_title')
```

lista dei cinema presenti nel
portale: ogni metodo è
accessibile dal template
tramite la variabile **view**.

approfondite qui: <http://plone.org/products/dexterity/documentation/manual/developer-manual>

realizzare una nuova vista 2

```
<html metal:use-macro="context/main_template/macros/master">
<body>
<metal:main fill-slot="main">
  <h1 tal:content="context/nome" />
  <p>Numero Sale:
    <em tal:content="context/numero_sale">#</em></p>
  <p>Orario Spettacoli:
    <em tal:content="context/orario_spettacoli">#</em></p>
  <h2>I nostri Cinema:</h2>
  <ul>
  <tal:block repeat="cinema view/lista_cinema">
    <li><a tal:attributes="href cinema/getURL"
      tal:content="cinema/nome" /></li>
  </tal:block>
  </ul>
</metal:main>
</body>
</html>
```

template renderizzato nella UI Plone

attributi accessibili direttamente sul contesto

accesso al metodo **lista_cinema** direttamente dalla variabile **view**.

template **view.pt** collocato nella cartella *cinema_templates*: agganciato alla classe per *convenzione*.

Non vi piacciono le convenzioni? usate ZCML!
<http://collective-docs.plone.org/en/latest/views/browserviews.html>

l'intorno della nostra vista



Benvenuto in Plone

creato da [admin](#) — ultima modifica 12/06/2012 19:16 — [Cronologia](#)

Congratulazioni! Hai completato con successo l'installazione di Plone.

[Disponibile anche in modalità presentazione...](#)

Se stai visualizzando questa pagina, invece del sito web che ti aspettavi, è perché il gestore di questo sito ha appena installato Plone. Non contattare il Team o la mailing list di Plone per questo.

approfondite qui:

<http://collective-docs.plone.org/en/latest/views/viewlets.html>

il `main_template` definisce l'intelaiatura dell'interfaccia.

Parte della *riusabilità* è ottenuta con le macro di ZPT.


ZTK ha permesso di componentizzare la UI tramite **Viewlet**.

UI estensibile con le viewlets

ViewletManager: plone.portalheader (plone.app.layout.viewlets.interfaces.IPortalHeader)

- Viewlet: plone.skip_links (0) ▼ Nascondi
- Viewlet: plone.personal_bar (1) ▲ ▼ Nascondi admin ▼
- Viewlet: plone.app.i18n.locales.languageselector (2) ▲ ▼ Nascondi
- Viewlet: plone.searchbox (3) ▲ ▼ Nascondi

Cerca nel sito

solo nella sezione corrente
- Viewlet: plone.logo (4) ▲ ▼ Nascondi
 
- Viewlet: plone.global_sections (5) ▲ Nascondi

Registrare una viewlet ad un Viewlet Manager per aggiungere "pezzi" di interfaccia senza "pestare i piedi" a nessuno.

La *posizione* e la *visibilità* delle viewlet è controllata da interfaccia web (@@manage-viewlets).

Le viewlet possono essere attivate tra l'altro da *behavior Dexterity*.

registriamo una viewlet

```
from five import grok
from plone.app.layout.viewlets.interfaces import IAboveContent
```

```
class MessageViewlet(grok.Viewlet):
    "mostriamo un messaggio"
    grok.name('example.messaging.MessageViewlet')
    grok.context(ICinema)
    grok.viewletmanager(IAboveContent)

    def update(self):
        self.message = IMessage(self.context)
```

nome del viewlet:
deve essere "unico"

viewlet attiva solo per
oggetti marcati ICinema

viewlet registrata per il Viewlet
Manager IAboveContent

L'adapter calcola il messaggio
in base a logiche definite
"altrove"

```
<div class="messageViewlet">
  <span tal:content="viewlet/message/subject" />
</div>
```

template **messageviewlet.pt** collocato
nella cartella *viewlet_templates*:
agganciato alla classe per *convenzione*.

ricapitolando..

- **non preoccupatevi dello storage**
(boilerplate.. migrazioni.. cambi di schema..)
- applicazioni **RESTful** molto più facili
- lasciate lavorare il **content framework**
- sfruttate i **servizi content-oriented**
- **Zope 2**, ma soprattutto **ZTK**
- molto molto altro "*sotto al cofano*"..



Grazie!
Domande?

Maurizio Delmonte - Abstract